

CoDynTrust: Robust Asynchronous Collaborative Perception via Dynamic Feature Trust Modulus

Yunjiang Xu¹, Lingzhi Li^{1*}, Jin Wang^{2*}, Benyuan Yang², Zhiwen Wu¹, Xinhong Chen³, Jianping Wang³

Abstract— Collaborative perception, fusing information from multiple agents, can extend perception range so as to improve perception performance. However, temporal asynchrony in real-world environments, caused by communication delays, clock misalignment, or sampling configuration differences, can lead to information mismatches. If this is not well handled, then the collaborative performance is patchy, and what's worse safety accidents may occur. To tackle this challenge, we propose CoDynTrust, an uncertainty-encoded asynchronous fusion perception framework that is robust to the information mismatches caused by temporal asynchrony. CoDynTrust generates dynamic feature trust modulus (DFTM) for each region of interest by modeling aleatoric and epistemic uncertainty as well as selectively suppressing or retaining single-vehicle features, thereby mitigating information mismatches. We then design a multi-scale fusion module to handle multi-scale feature maps processed by DFTM. Compared to existing works that also consider asynchronous collaborative perception, CoDynTrust combats various low-quality information in temporally asynchronous scenarios and allows uncertainty to be propagated to downstream tasks such as planning and control. Experimental results demonstrate that CoDynTrust significantly reduces performance degradation caused by temporal asynchrony across multiple datasets, achieving state-of-the-art detection performance even with temporal asynchrony. The code is available at <https://github.com/CrazyShout/CoDynTrust>.

I. INTRODUCTION

With the rapid development of autonomous driving, 3D object detection, a core technology for environmental perception, has gained widespread attention [1], [2], [3]. Its goal is to accurately identify and locate objects using sensor data from LiDAR and cameras. However, single-vehicle sensors have limited coverage, especially in complex urban environments with occlusion and reduced visibility [4], [5]. This makes comprehensive perception difficult. As an alternative, collaborative perception arises, which can significantly improve perception range and accuracy by sharing and processing data among multiple agents via vehicle-to-everything (V2X) technology [6], [7], [8].

Although collaborative perception mitigates the limitations of single-vehicle perception, it still faces practical issues such as positioning errors [9], bandwidth limitations [10], and latency [11]. Among them, temporal asynchrony is one of

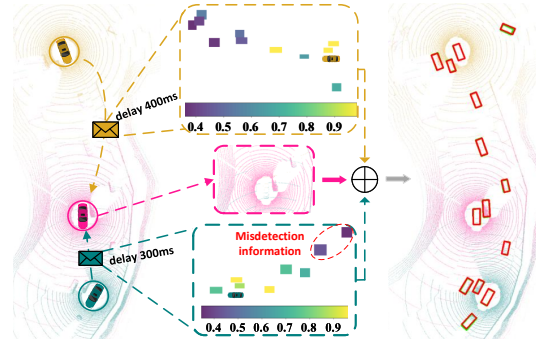


Fig. 1. Based on DFTM, CoDynTrust can mitigate error propagation amplified by temporal asynchrony, effectively suppressing low-quality information and enhancing detection robustness.

the most difficult challenges since it can lead to mismatches in sensor data, thus causing collisions and conflicts between different vehicles. There exist several methods that use delay compensation strategies to address temporal asynchrony [11], [12], [13], [14], [15]. However, they are limited to processing single-frame data and do not fully utilize historical information. Moreover, their performance is limited in high-latency or complex scenarios. In reality, asynchronous communication is inevitable due to unsynchronized sampling, different processing speed, or network congestion. This poses threats to the accuracy and real-time performance of perception.

To address the temporal asynchrony in collaborative perception, we propose CoDynTrust, as illustrated in Fig. 1. CoDynTrust evaluates the dynamic feature trust modulus (DFTM) of each region of interest (ROI) so as to tackle error propagation in the ROI generator. Furthermore, we adopt a simple linear extrapolation method instead of a learned motion prediction model so as to ensure reasonable vehicle movement within common latency ranges (0s–0.5s). Then, we design an adaptive fusion method to enhance system performance. Finally, we conduct extensive experiments on real and simulated datasets for LiDAR-based 3D object detection tasks, including DAIR-V2X [16], V2XSet [13], and OPV2V [17]. Experimental results show that: i) DFTM effectively addresses detection quality inconsistencies caused by noise, model flaws, and asynchronous delays. ii) CoDynTrust consistently outperforms previous state-of-the-art methods in delay scenarios. In summary, the main contributions of this paper are as follows:

- We propose CoDynTrust, a novel LiDAR-based 3D detection framework for robust asynchronous multi-agent perception, addressing detection errors from noise,

¹School of Computer Science and Technology, Soochow University, Suzhou 215006, China.

²School of Future Science and Engineering, Soochow University, Suzhou 215299, China.

³Computer Science of Department, City University of Hong Kong.

Emails: yjxu95@stu.suda.edu.cn, {lilingzhi, wjin1985, byyang}@suda.edu.cn, zwwu@stu.suda.edu.cn, {xinhong.chen, jianwang}@cityu.edu.hk

*Corresponding author.

model defects, and worsened by delays.

- We introduce DFTM, an uncertainty-based method estimating each agent's ROI trustworthiness. Combined with adaptive fusion, DFTM adjusts feature fusion and can be propagated to downstream tasks (e.g., planning, control) to enhance system robustness.
- We validate CoDynTrust via extensive experiments. Results show that CoDynTrust significantly mitigates the performance degradation caused by temporal asynchrony and improves robustness in delay scenarios.

II. RELATED WORKS

A. Collaborative Perception

Collaborative perception addresses the challenges of limited perception range and occlusion in single-vehicle autonomous driving by enabling information exchange between agents. High-quality datasets like DAIR-V2X [16], V2XSet [13], OPV2V [17], and V2X-Sim [18] have been given, and notable methods are proposed, such as DiscoNet [19] using a teacher-student distillation framework to extract more information during training, V2X-ViT [13] the first heterogeneous Transformer for V2X perception, CoAlign [9] which enhances consistency between agents using pose graph modeling, and CoBEVFlow [15] which mitigates delays through motion prediction.

B. Temporal Asynchrony Issue in Collaborative Perception

Temporal asynchrony is unavoidable in real-world environments, exacerbating errors caused by noise and model deficiencies. Existing solutions like V2VNet [12] employ CNNs for delay compensation, while V2X-ViT [13] mitigates spatio-temporal distortions via position encoding. However, these methods insufficiently leverage historical data. Though SyncNet [11] applies Conv-LSTM [20] for delay handling, its RNN-based feature generation struggles with noise and irregular delays. FFNet [14] processes irregular multi-frame data but introduces generation noise and infrastructure dependencies. CoBEVFlow [15] demands high-quality ROI generation, and its three-stage training risks error propagation from delays. To overcome these limitations, we propose an optimized asynchronous collaborative perception framework.

C. Uncertainty Quantification

Uncertainty quantification critically evaluates prediction reliability through two categories: aleatoric (data noise irreducible with data volume) and epistemic (reducible model limitations). Aleatoric uncertainty is captured through direct modeling (DM) [21] with variance prediction outputs; epistemic uncertainty estimated via Monte Carlo dropout [22] approximating Bayesian inference or deep ensembles [23]. In autonomous driving, this enhances safety in perception [24], [25] and decision-making. Current collaborative perception research predominantly applies uncertainty to downstream tasks like planning and control [26], [27], neglecting specific challenge resolution. In CoDynTrust, we quantify both uncertainties using DM and MC dropout to guide asynchronous

collaborative perception, filtering out noise and low-quality data, and propagating uncertainty to tasks such as planning and control.

III. PROBLEM FORMULATION

Consider a scenario with N agents, where each agent can exchange information with others and store two frames of historical data from others. For the n -th agent, let $\mathbf{X}_n^{t_n^i}$ and $\mathbf{Y}_n^{t_n^i}$ respectively represent the perception observation and supervision at time t_n^i . Here, t_n^i is the timestamp of agent n at the i -th moment. $\mathcal{P}_{m \rightarrow n}^{t_m^j}$ denotes the collaborative message sent from agent m to agent n at t_m^j . Temporal asynchrony includes: i) timestamp misalignment between agents, $t_m^j \neq t_n^i$, and ii) irregular time intervals between two frames of received messages from other agents, i.e., $t_m^j - t_m^{j-1}$. Therefore, the task of asynchronous collaborative perception can be formulated as:

$$\begin{aligned} & \arg \max_{\theta, \mathcal{P}} \sum_{n=1}^N g(\hat{\mathbf{Y}}_n^{t_n^i}, \mathbf{Y}_n^{t_n^i}), \\ & \text{subject to } \hat{\mathbf{Y}}_n^{t_n^i} = \Phi_{\theta} \left(\mathbf{X}_n^{t_n^i}, \left\{ \mathcal{P}_{m \rightarrow n}^{t_m^j}, \mathcal{P}_{m \rightarrow n}^{t_m^{j-1}} \right\}_{m=1}^N \right), \end{aligned} \quad (1)$$

In this formula, $g(\cdot, \cdot)$ represents the perception evaluation metric, and $\hat{\mathbf{Y}}_n^{t_n^i}$ denotes the n -th agent's detection result at time t_n^i . The collaborative perception network Φ is parameterized by θ . When timestamps are synchronized and frame intervals are uniform (i.e., $t_m^j = t_n^i$ and $t_n^i - t_n^{i-1}$ is constant), the problem simplifies to the ideal delay-free case of uniform. If $t_m^j \neq t_n^i$ but intervals remain uniform, the setup resembles SyncNet's scenario.

IV. ASYNCHRONY-ROBUST COLLABORATIVE PERCEPTION

A. Overall Architecture

Temporal asynchrony means that agents perceive the same object at different times. Existing methods mitigate them by: i) using a pre-trained ROI generator after feature extraction, ii) feeding multi-frame ROIs into a motion prediction module to generate birds' eye view (BEV) flow map, and iii) warping asynchronous feature with the BEV flow map before fusion. However, such strategy has two issues: i) over-reliance on the ROI generator, leading to variable ROI quality due to noise and model limitations; and ii) errors accumulate in both ROI and BEV flow generation, worsened by irregular delays. CoDynTrust addresses these issues by: i) parallel uncertainty quantification for each ROI, and ii) using linear extrapolation for motion prediction and averaging uncertainty across frames to generate DFTM. By doing so, we can filter noise and retain high-quality features. Furthermore, an adaptive fusion further enhances feature integration. Formally,

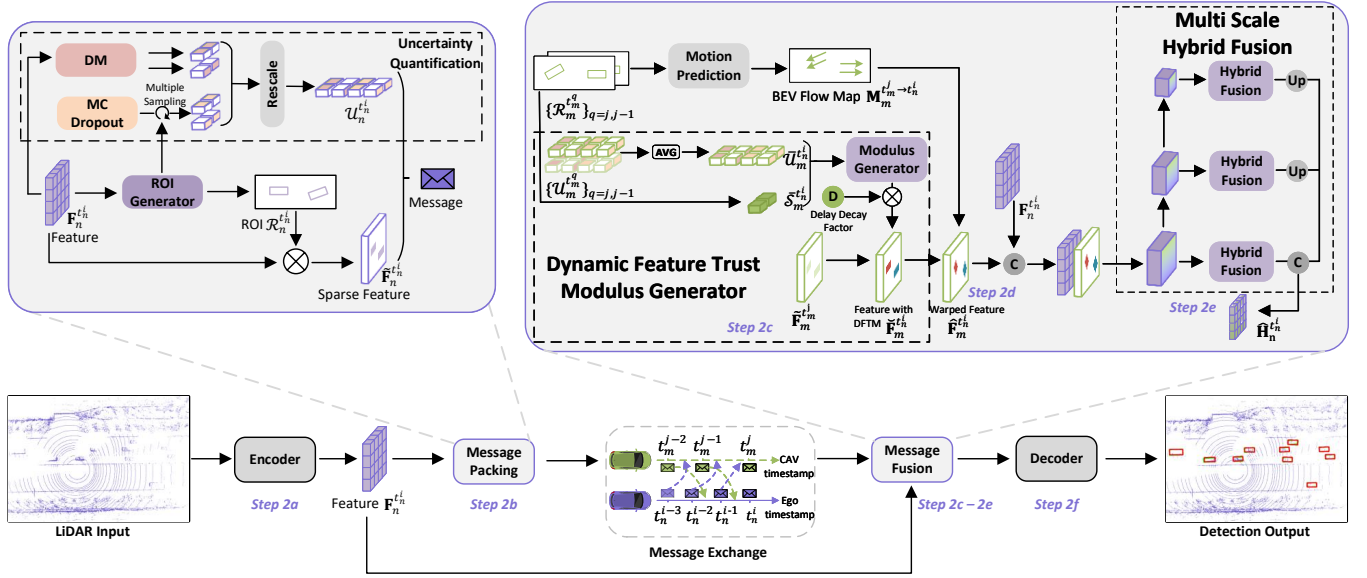


Fig. 2. System overview. The message packing process prepares ROI, uncertainty, and sparse features for efficient communication and BEV flow map generation. Message fusion generate Dynamic Feature Trust Modulus and scatters it back to the sparse feature map, while the BEV map is generated and used for motion compensation. Finally, multi-scale Hybrid Fusion is applied to fuse feature maps from all agents.

CoDynTrust can be expressed as:

$$\mathbf{F}_n^{t_n^i} = f_{\text{enc}}(\mathbf{X}_n^{t_n^i}), \quad (2a)$$

$$\mathcal{R}_n^{t_n^i}, \mathcal{U}_n^{t_n^i}, \tilde{\mathbf{F}}_n^{t_n^i} = f_{\text{roi_gen}}(\mathbf{F}_n^{t_n^i}), \quad (2b)$$

$$\{\mathbf{M}_m, \mathbf{D}_m\}^{\Delta t} = f_{\text{flow_gen}}(t_n^i, \mathbf{R}_m, \mathbf{U}_m), \quad (2c)$$

$$\hat{\mathbf{F}}_m^{t_n^i} = f_{\text{warp_dtfm}}(\tilde{\mathbf{F}}_m^{t_n^i}, \{\mathbf{M}_m, \mathbf{D}_m\}^{\Delta t}), \quad (2d)$$

$$\hat{\mathbf{H}}_n^{t_n^i} = f_{\text{agg}}(\tilde{\mathbf{F}}_n^{t_n^i}, \{\hat{\mathbf{F}}_m^{t_n^i}\}_{m \in \mathcal{N}_n}), \quad (2e)$$

$$\hat{\mathbf{Y}}_n^{t_n^i} = f_{\text{dec}}(\hat{\mathbf{H}}_n^{t_n^i}), \quad (2f)$$

Here, $\mathbf{F}_n^{t_n^i} \in \mathbb{R}^{H \times W \times D}$ is n -th agent's BEV feature at timestamp t_n^i , where H and W are its size and D is the channels. $\mathcal{R}_n^{t_n^i}$ is the set of ROIs, $\mathcal{U}_n^{t_n^i}$ is the corresponding uncertainty, and $\tilde{\mathbf{F}}_n^{t_n^i}$ is the sparse version of $\mathbf{F}_n^{t_n^i}$ containing features within $\mathcal{R}_n^{t_n^i}$. $\{\mathbf{M}_m, \mathbf{D}_m\}^{\Delta t}$ refers to the BEV flow map $\mathbf{M}_m^{t_m^j \rightarrow t_n^i} \in \mathbb{R}^{H \times W \times 2}$, reflecting grid cells movement from t_m^j to t_n^i . $\mathbf{D}_m^{t_m^j \rightarrow t_n^i} \in \mathbb{R}^{o \times 1}$ represents the trust modulus of each ROI, where o is the number of matched ROI pairs. $\hat{\mathbf{F}}_m^{t_n^i}$ is the re-aligned feature after motion compensation and DFTM application. $\hat{\mathbf{H}}_n^{t_n^i}$ is the aggregated feature from all agents, where $\mathcal{N}_n$ denotes the collaborative neighbors of agent n . $\hat{\mathbf{Y}}_n^{t_n^i}$ is the system's output.

The whole procedure of our approach is given in Fig. 2. Specifically, Step 2a extracts BEV features from observation data. Step 2b generates ROIs and evaluates uncertainty, and then agents exchange messages including $\mathcal{U}_n^{t_n^i}$, $\mathcal{R}_n^{t_n^i}$, and $\tilde{\mathbf{F}}_n^{t_n^i}$. Step 2c generates BEV flow map. Step 2d applies the BEV flow map to warp asynchronous features and apply DFTM so as to obtain re-aligned feature map. Step 2e aggregates features from all agents. Step 2f outputs the final perception result. Note that, steps 2a and 2b run without communication,

while communication begins in Steps 2c-2f. Next, we explain the main steps and contents of CoDynTrust.

B. ROI Generation with Uncertainty Quantification

Uncertainty Quantification. For agent m at timestamp t_m^j with BEV feature $\mathbf{F}_m^{t_m^j}$, CoDynTrust quantifies aleatoric and epistemic uncertainty. Each type of uncertainty corresponds to classification and regression uncertainty. Let $\mathbf{P}_{\text{DM}}(\cdot)$ and $\mathbf{P}_{\text{MCD}}(\cdot)$ represent the processes of DM and MC Dropout, respectively. The quantification is:

$$\{u_{\text{ale}}^{\text{cls}}, u_{\text{ale}}^{\text{reg}}\}_m^{t_m^j} = \mathbf{P}_{\text{DM}}(\mathbf{F}_m^{t_m^j}) \quad (3a)$$

$$\{u_{\text{epi}}^{\text{cls}}, u_{\text{epi}}^{\text{reg}}\}_m^{t_m^j} = \mathbf{P}_{\text{MCD}}(\mathbf{X}_m^{t_m^j}, \theta_{\text{single}}, T) \quad (3b)$$

Since we evaluate ROI uncertainty, the regression aleatoric uncertainty is evaluated by modeling the bounding box center (x, y) as Gaussian random variable. Considering the periodicity of yaw angle α , it is modeled α with von Mises distribution. The regression head predicts the mean, and an additional head predicts the log variance (for x and y) and the inverse concentration parameter (for α) to capture uncertainty.

Similarly, the classification aleatoric uncertainty is modeled by treating the logit of the classification head as Gaussian random variable. An additional head predicts the variance to quantify classification noise. Since it is challenging to directly optimize the Gaussian parameters [28], we apply the reparameterization trick [29]. Specifically, we express the random variable as a differentiable function by introducing standard normal noise, scaled and shifted for smooth gradient propagation. This approach effectively learns classification uncertainty while maintaining stability.

We use MC Dropout to quantify epistemic uncertainty. In equation (3b), $\mathbf{X}_m^{t_m^j}$ is the raw observation of agent

m at timestamp t_m^j , θ_{single} is the single-vehicle detection model parameter, and T is the number of MC Dropout inferences. Since dropout is used only in the decoder, during inference, step 2b is executed T times. The mean of the T results is taken as the final output, with classification entropy and regression variance representing the epistemic uncertainties. Finally, the raw uncertainty set for agent m at t_m^j is denoted as: $(\mathcal{U}_m^{t_m^j})_{raw} = \{u_{ale}^{cls}, u_{ale}^{reg}, u_{epi}^{cls}, u_{epi}^{reg}\}_{m}^{t_m^j}$. Due to the differences in evaluation methods and unitless properties of uncertainty, the classification or regression uncertainties cannot be directly compared. Therefore, we perform a rescaling operation, denoted as $\mathbf{P}_{rescale}(\cdot)$, with the process: $(\mathcal{U}_m^{t_m^j})_{dense} = \mathbf{P}_{rescale}((\mathcal{U}_m^{t_m^j})_{raw})$. Specifically, for classification uncertainty, we use the classification deviation ratio [24] to unify the scale:

$$u^{cls} = \frac{\frac{\mu_u}{\mu_u + ReLU(u_{raw}^{cls} - \mu_u - \sigma_u)}}{\frac{\mu_s}{\mu_s + ReLU(-(s_c - \mu_s - \sigma_s))}}, \quad (4)$$

Here, u_{raw}^{cls} is one type of classification uncertainty, and s_c is the confidence score. μ_u , σ_u , μ_s , and σ_s are the mean and variance of raw classification uncertainty and confidence scores for positive samples in the test set. Compared to raw classification uncertainty, the classification deviation ratio provides more information and amplifies the difference between positive and negative samples. Finally, regression uncertainty is scaled by multiplying the uncertainty of the bounding box center (x, y) by the box's diagonal and applying Z-Score normalization for a unified scale.

ROI Set Generation. The BEV feature from step 2a is fed into the ROI generator to generate the ROI: $\mathbf{O}_m^{t_m^j} = \Phi_{roi-gen}^T(\mathbf{F}_m^{t_m^j}) \in \mathbb{R}^{H \times W \times 8}$, where $\Phi_{roi-gen}^T(\cdot)$ is the ROI generation network, and T is the number of MC Dropout inferences. Each element $(\mathbf{O}_m^{t_m^j})_{h,w} = (c, x, y, z, d_x, d_y, d_z, \alpha)$ is the mean of the T detection results for class confidence, position, size, and orientation of an ROI. Post-processing then generates detection boxes, projected into BEV space as the ROI set $\mathcal{R}_m^{t_m^j}$, with each element $(c, x, y, d_x, d_y, \alpha)$. Finally, the uncertainty set $\mathcal{U}_m^{t_m^j}$ corresponding to the ROI set is indexed.

Sparse Feature Map Generation. A binary mask $\mathbf{H} \in \mathbb{R}^{H \times W}$ is generated from the ROI set, where regions inside are 1 and outside are 0. Applying this mask to the feature map from step 2a, we have $\tilde{\mathbf{F}}_m^{t_m^j} = \mathbf{F}_m^{t_m^j} \odot \mathbf{H}$, retaining only ROI features. Agent m then packages $\mathcal{U}_m^{t_m^j}$, $\mathcal{R}_m^{t_m^j}$, and $\tilde{\mathbf{F}}_m^{t_m^j}$, and sends them to other agents.

C. Flow Generation with DFTM

After communication, collaborative nodes receive information from others, often with irregular time delays. To address this, we require: i) efficient and interpretable motion prediction to avoid error amplification; and ii) dynamic feature trust modulus to evaluate historical data to reduce noise. Step 2c can achieve so. Specifically, a BEV flow map is generated to record displacements and calibrate the

received features. Unlike learning-based models that require multiple frames, CoDynTrust only needs two. Each frame's ROIs are matched as detection proposals using an efficient algorithm [15], followed by motion prediction generating the BEV flow map.

CoDynTrust uses linear extrapolation for motion prediction. For the r -th ROI of agent m at timestamp t_m^j , the motion attributes are extracted, i.e., $\{\mathbf{P}_r^{t_m^j}, \mathbf{P}_r^{t_m^{j-1}}\}$, where $\mathbf{P}_r^{t_m^j} = (x_r^{t_m^j}, y_r^{t_m^j}, \alpha_r^{t_m^j})$. The ROI velocity is $v_r = (\mathbf{P}_r^{t_m^j} - \mathbf{P}_r^{t_m^{j-1}})/(t_m^j - t_m^{j-1})$, and the motion displacement is $\Delta d_r = v_r \times (t_n^i - t_m^j)$. Δd_r is then used to create the BEV flow map $\mathbf{M}_m^{t_m^j \rightarrow t_n^i}$, storing motion mappings in two axes for each grid in BEV space within the ROI area, while other regions retain an identity mapping.

Then, to dynamically generate the feature trust modulus, CoDynTrust combines the confidence and uncertainty of all ROIs from historical frames: $\mathbf{D}_m^{t_m^j \rightarrow t_n^i} = \Phi_{dftm}(\bar{\mathbf{S}}_m^{t_n^i}, \bar{\mathbf{U}}_m^{t_n^i}) \times d_f$, where $\Phi_{dftm}(\cdot)$ is the DFTM generation network, and $\bar{\mathbf{S}}_m^{t_n^i}$ and $\bar{\mathbf{U}}_m^{t_n^i}$ are the averages of confidence and uncertainty, respectively. Since the previous matching filters out extreme outliers, a simple average works well. d_f is the delay decay factor, resisting noise under high delays. $\Phi_{dftm}(\cdot)$ uses residual blocks to produce the feature trust modulus between 0 and 1. After compensating and re-estimating ROI confidence and uncertainty, a raw modulus is obtained. It's then multiplied by $d_f = \exp(-k \times (t_n^i - t_m^j))$, where $k = 0.02$, to further adjust for low-reliability ROIs under high delays. The final output is $\mathbf{D}_m^{t_m^j \rightarrow t_n^i} \in \mathbb{R}^{o \times 1}$.

Feature Warp Guided by DFTM. In step 2d, the BEV flow map and DFTM are applied to the feature map. Each sparse feature grid shifts based on the flow map, while DFTM is scattered back into the feature map and scales each ROI's feature. This step is denoted as $\mathbf{P}_{scatter}(\cdot)$, and the overall transformation is: $\hat{\mathbf{F}}_m^{t_n^i}[h, w] = \mathbf{P}_{scatter}(\mathbf{D}_m^{t_m^j \rightarrow t_n^i}) \odot \tilde{\mathbf{F}}_m^{t_m^j}[h + \mathbf{M}_m^{t_m^j \rightarrow t_n^i}[h, w, 0], w + \mathbf{M}_m^{t_m^j \rightarrow t_n^i}[h, w, 1]]$. Through this transformation, the feature maps of non-ego agents achieve asynchronous compensation.

D. Hybrid Fusion

In step 2e, i.e., hybrid fusion as shown in Fig. 3, fuses features from other agents at multiple scales to adapt to objects of varying sizes. Since the features processed by DFTM exhibit sparsity and scaling, traditional methods struggle with effective fusion. Hybrid fusion concatenates features applying both MAXOUT [30] and AVGOUT operations to high-light key features and retain multi-agent perspectives. The results are concatenated and passed through convolutional layers and a sigmoid function to generate spatial weights, which are used to weight and sum the MAXOUT and AVGOUT outputs, forming the fused feature map. Global average pooling and a linear layer produce channel weights that further refine the feature map. The final output $\hat{\mathbf{H}}_n^{t_n^i}$ integrates spatial and channel information. Furthermore, for smooth gradient propagation, we use Mish [31] activation throughout.

TABLE I

PERFORMANCE COMPARISON OF CoDynTrust AND BASELINE METHODS AT EXPECTED DELAYS FROM 0MS TO 500MS ON THREE DATASETS.

Dataset Expectation of interval (ms)	DAIR-V2X			V2XSet			OPV2V(Culver)		
	0	300	500	0	300	500	0	300	500
Model / Metric	AP@0.5 ↑								
Single	0.6349			0.6515			0.6427		
Late Fusion	0.6043	0.5401	0.5045	0.7876	0.4671	0.4112	0.8049	0.6702	0.6088
V2VNet [12]	0.7395	0.6572	0.6298	0.8691	0.5766	0.4990	0.8133	0.7332	0.6909
V2X-ViT [13]	0.7820	0.7202	0.7012	0.8785	0.6549	0.5535	0.8435	0.7788	0.7397
Where2comm [10]	0.7921	0.7103	0.6861	0.8918	0.6767	0.6131	0.8695	0.8014	0.7677
Where2comm+SyncNet [11]	0.7921	0.7324	0.7103	0.8918	0.8081	0.7773	0.8695	0.8106	0.7892
CoBEVFlow [15]	0.7751	0.7419	0.7316	0.8632	0.8542	0.8388	0.8370	0.8217	0.8186
Ours	0.7922	0.7481	0.7357	0.8780	0.8644	0.8475	0.8646	0.8566	0.8532
Model / Metric	AP@0.7 ↑								
Single	0.4962			0.5204			0.5213		
Late Fusion	0.3746	0.3226	0.3027	0.6801	0.3239	0.3152	0.6809	0.5031	0.4719
V2VNet [12]	0.5835	0.4982	0.4945	0.7675	0.3981	0.3715	0.6398	0.4864	0.4578
V2X-ViT [13]	0.6590	0.6046	0.5973	0.7170	0.4618	0.4179	0.6834	0.5990	0.5689
Where2comm [10]	0.6920	0.6066	0.5952	0.7966	0.5026	0.4854	0.7858	0.6638	0.6393
Where2comm+SyncNet [11]	0.6920	0.6246	0.6121	0.7966	0.6720	0.6511	0.7858	0.7057	0.6906
CoBEVFlow [15]	0.6627	0.6264	0.6207	0.8111	0.7638	0.7340	0.7832	0.7400	0.7315
Ours	0.7001	0.6431	0.6368	0.8356	0.7869	0.7582	0.8194	0.7952	0.7833

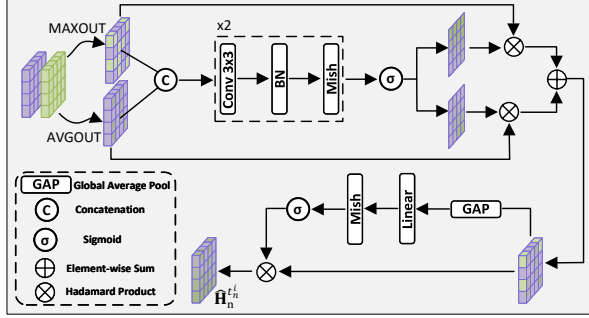


Fig. 3. Overall Structure of Hybrid Fusion.

V. EXPERIMENTAL RESULTS

We conducted experiments on three benchmark datasets. All experiments focus on LiDAR 3D object detection.

A. Dataset

DAIR-V2X [16] is a real-world collaborative perception dataset with two agents (a vehicle and an infrastructure) per sample. Both capture image and point cloud data with 3D annotations. The LiDAR range is $201.6m \times 80m$ with a 10Hz sampling frequency. We applied additional annotations [9] to cover objects outside the camera's field of view, achieving 360° detection coverage. **V2XSet** [13] is a V2X simulation dataset generated using Carla [32] and OpenCDA [33], containing 73 collaborative scenarios with 11,447 frames of point cloud data. Each scenario includes 2 to 4 vehicles and infrastructure agents. The LiDAR range is $281.6m \times 80m$, with standard splits for training, validation, and testing. **OPV2V** [17] is a vehicle-to-vehicle collaborative perception dataset simulated by OpenCDA and Carla. It includes 70 scenarios and 11,464 frames with over 232,913 3D annotations. The Culver City Digital Twin test set,

which reflects real-world conditions, is used to evaluate generalization performance.

B. Impelment Details

CoDynTrust's training has two stages: pre-training an ROI generator and training the fusion detector with frozen ROI generator parameters. Both use the same structure without sharing parameters. We use PointPillar [1] as the encoder with a grid size of $(0.4m, 0.4m)$. We apply Focal Loss [34] for classification, weighted smooth L1 Loss for regression, and negative log-likelihood loss for uncertainty modeling. The AdamW [35] optimizer is used with learning rates of 0.001 for the ROI generator and 0.002 for the fusion detector. All experiments run on an NVIDIA A100 GPU.

C. Quantitative Results

Benchmark Comparison. To evaluate CoDynTrust's performance under temporal asynchrony, we compared it with multiple models, both with and without asynchronous robustness designs. Referencing CoBEVFlow's setup, we applied binomially sampled frame intervals to create irregular time gaps. Table I shows the AP results at IoU thresholds of 0.5 and 0.7 on the DAIR-V2X, V2XSet, and OPV2V datasets. CoDynTrust consistently outperformed in nearly all scenarios, showing strong robustness to temporal asynchrony.

Trade-off Between Detection Performance and Communication Cost. CoDynTrust transmits uncertainty, ROIs and sparse features, resulting in low communication costs and being bandwidth-friendly. Fig. 4 shows the trade-off between bandwidth and performance (AP@0.7) under expected 300ms temporal asynchrony for CoDynTrust and other methods. Experiments on DAIR-V2X and V2XSet show: i) CoDynTrust consistently outperforms others with the same bandwidth. ii) On V2XSet, CoDynTrust reached

its optimal performance with lower bandwidth requirements than Where2Comm.

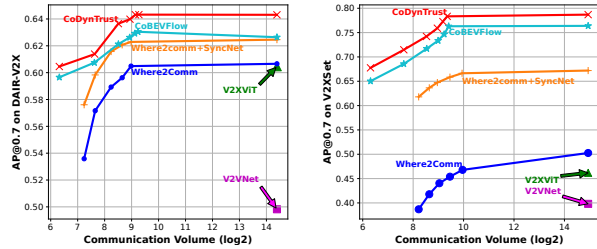


Fig. 4. Trade-off between detection performance (AP@0.7) and communication bandwidth under expected 300ms delay on DAIR-V2X (left) and V2XSet (right) datasets.

Robustness to Pose Errors. To evaluate system robustness, we test performance under temporal asynchrony and pose errors. Pose errors are simulated during inference by adding Gaussian noise to the 2D center and yaw angle, with means of $0m/0^\circ$ and standard deviations ranging from $0m - 0.5m$ and $0^\circ - 0.5^\circ$. The expected frame interval is set to 300ms for temporal asynchrony. Table II compares SOTA models' robustness on the DAIR-V2X dataset with pose errors. CoDynTrust maintained top performance even with both temporal asynchrony and pose errors, outperforming CoBEVFlow by 2.86% in AP@0.7 at $0.4m/0.4^\circ$ noise.

TABLE II

DETECTION PERFORMANCE ON THE DAIR-V2X DATASET WITH GAUSSIAN-DISTRIBUTED POSE NOISE DURING TESTING.

Noise Level $\sigma_t/\sigma_r(m/o)$	0.0/0.0	0.1/0.1	0.2/0.2	0.3/0.3	0.4/0.4
Model / Metric	AP@0.5 $\uparrow$				
V2VNet	0.6572	0.6476	0.6336	0.6144	0.5974
V2X-ViT	0.7202	0.7098	0.6954	0.6733	0.6614
Where2comm	0.7103	0.7087	0.7003	0.6904	0.6793
Where2comm+SyncNet	0.7324	0.7321	0.7238	0.7100	0.6953
CoBEVFlow	0.7419	0.7397	0.7347	0.7237	0.7145
Ours	0.7481	0.7411	0.7378	0.7289	0.7164
Model / Metric	AP@0.7 $\uparrow$				
V2VNet	0.4982	0.4879	0.4705	0.4624	0.4512
V2X-ViT	0.6046	0.5916	0.5758	0.5653	0.5611
Where2comm	0.6066	0.6020	0.5875	0.5771	0.5703
Where2comm+SyncNet	0.6246	0.6178	0.6032	0.5941	0.5877
CoBEVFlow	0.6264	0.6217	0.6112	0.6042	0.5954
Ours	0.6431	0.6366	0.6252	0.6209	0.6124

D. Qualitative Results

Fig. 5 shows the detection results of CoDynTrust and CoBEVFlow on V2XSet dataset, with a expected time interval of 300ms. Red boxes indicate detection results, while green boxes represent the ground truth. We compare how the two models handle features within the ROI. In complex temporal asynchrony scenarios, CoBEVFlow struggles to distinguish individual vehicle detections, leading to the sharing of low-quality detection information harms overall performance. In contrast, CoDynTrust retains reliable detections, suppressing noise and ensuring robustness.

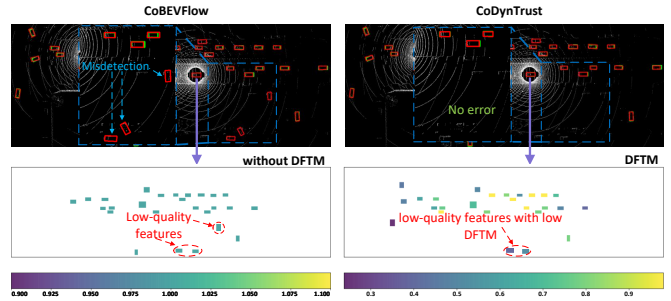


Fig. 5. Visualization of CoBEVFlow and CoDynTrust detection results on V2XSet, with an expectation of a 300ms time interval. CoDynTrust shows better detection quality compared to CoBEVFlow. Red boxes indicate detection results, while green boxes represent ground truth.

E. Ablation Studies

We conducted ablation experiments on DAIR-V2X dataset to evaluate the effectiveness of each component. Table III shows the results, highlighting: i) Isolated uncertainty quantification yields marginal gains; ii) Combining uncertainty quantification with DFTM results in a significant performance boost; iii) The Hybrid Fusion mechanism, specifically designed for DFTM, further optimizes feature fusion and provides an additional performance gain.

TABLE III

ABLATION STUDY RESULTS ON DAIR-V2X, **DFTM**: DYNAMIC FEATURE TRUST MODULUS.

Modules			AP@0.5 $\uparrow$	AP@0.7 $\uparrow$
Uncertainty	DFTM	Hybrid Fusion		
✓			0.7419	0.6264
✓	✓		0.7420	0.6265
✓		✓	0.7472	0.6334
✓	✓	✓	0.7481	0.6431

VI. CONCLUSION

We proposed CoDynTrust to tackle temporal asynchrony in collaborative perception. Using DFTM for ROI reliability and linear extrapolation for motion prediction, CoDynTrust enhances interpretability and feature fusion. Experiments show it consistently outperforms existing methods in asynchronous settings and demonstrates strong robust. Additionally, DFTM can also be propagated to downstream tasks, enhancing safety and reliability in autonomous driving. Future work will focus on optimizing DFTM generation and explore its further application in downstream modules.

VII. ACKNOWLEDGE

This work was supported in part by the National Natural Science Foundation of China (62072321), the Six Talent Peak Project of Jiangsu Province (XYDXX-084), the Science and Technology Program of Jiangsu Province (BZ2024062), the Natural Science Foundation of the Jiangsu Higher Education Institutions of China (22KJA520007), Suzhou Planning Project of Science and Technology (2023ss03), Hong Kong Research Grant Council Under GRF (11210622).

REFERENCES

- [1] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "Pointpillars: Fast encoders for object detection from point clouds," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 12 697–12 705.
- [2] S. Chen, B. Liu, C. Feng, C. Vallespi-Gonzalez, and C. Wellington, "3d point cloud processing and learning for autonomous driving: Impacting map creation, localization, and perception," *IEEE Signal Processing Magazine*, vol. 38, no. 1, pp. 68–86, 2020.
- [3] G. Zhang, J. Chen, G. Gao, J. Li, S. Liu, and X. Hu, "Safednet: A simple and effective network for fully sparse 3d object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024, pp. 14 477–14 486.
- [4] Z. Yuan, X. Song, L. Bai, Z. Wang, and W. Ouyang, "Temporal-channel transformer for 3d lidar-based video object detection for autonomous driving," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 4, pp. 2068–2078, 2021.
- [5] Z. Zhang and J. F. Fisac, "Safe occlusion-aware autonomous driving via game-theoretic active perception," *arXiv preprint arXiv:2105.08169*, 2021.
- [6] S. Fan, H. Yu, W. Yang, J. Yuan, and Z. Nie, "Quest: Query stream for practical cooperative perception," in *Proceedings of the International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 18 436–18 442.
- [7] W. Su, L. Chen, Y. Bai, X. Lin, G. Li, Z. Qu, and P. Zhou, "What makes good collaborative views? contrastive mutual information maximization for multi-agent perception," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 16, 2024, pp. 17 550–17 558.
- [8] K. Yang, D. Yang, J. Zhang, M. Li, Y. Liu, J. Liu, H. Wang, P. Sun, and L. Song, "Spatio-temporal domain awareness for multi-agent collaborative perception," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023, pp. 23 383–23 392.
- [9] Y. Lu, Q. Li, B. Liu, M. Dianati, C. Feng, S. Chen, and Y. Wang, "Robust collaborative 3d object detection in presence of pose errors," in *Proceedings of the International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 4812–4818.
- [10] Y. Hu, S. Fang, Z. Lei, Y. Zhong, and S. Chen, "Where2comm: Communication-efficient collaborative perception via spatial confidence maps," in *Advances in neural information processing systems (NeurIPS)*, vol. 35, 2022, pp. 4874–4886.
- [11] Z. Lei, S. Ren, Y. Hu, W. Zhang, and S. Chen, "Latency-aware collaborative perception," in *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer, 2022, pp. 316–332.
- [12] T.-H. Wang, S. Manivasagam, M. Liang, B. Yang, W. Zeng, and R. Urtasun, "V2vnet: Vehicle-to-vehicle communication for joint perception and prediction," in *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer, 2020, pp. 605–621.
- [13] R. Xu, H. Xiang, Z. Tu, X. Xia, M.-H. Yang, and J. Ma, "V2x-vit: Vehicle-to-everything cooperative perception with vision transformer," in *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer, 2022, pp. 107–124.
- [14] H. Yu, Y. Tang, E. Xie, J. Mao, P. Luo, and Z. Nie, "Flow-based feature fusion for vehicle-infrastructure cooperative 3d object detection," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 36, 2023, pp. 34 493–34 503.
- [15] S. Wei, Y. Wei, Y. Hu, Y. Lu, Y. Zhong, S. Chen, and Y. Zhang, "Asynchrony-robust collaborative perception via bird's eye view flow," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 36, 2023, pp. 28 462–28 477.
- [16] H. Yu, Y. Luo, M. Shu, Y. Huo, Z. Yang, Y. Shi, Z. Guo, H. Li, X. Hu, J. Yuan, and Z. Nie, "Dair-v2x: A large-scale dataset for vehicle-infrastructure cooperative 3d object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 21 329–21 338.
- [17] R. Xu, H. Xiang, X. Xia, J. L. Xu Han, and J. Ma, "Opv2v: An open benchmark dataset and fusion pipeline for perception with vehicle-to-vehicle communication," in *Proceedings of the International Conference on Robotics and Automation (ICRA)*, 2022, pp. 2583–2589.
- [18] Y. Li, D. Ma, Z. An, Z. Wang, Y. Zhong, S. Chen, and C. Feng, "V2x-sim: Multi-agent collaborative perception dataset and benchmark for autonomous driving," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 10914–10921, 2022.
- [19] Y. Li, S. Ren, P. Wu, S. Chen, C. Feng, and W. Zhang, "Learning distilled collaboration graph for multi-agent perception," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 34, 2021, pp. 29 541–29 552.
- [20] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-c. Woo, "Convolutional lstm network: A machine learning approach for precipitation nowcasting," in *Advances in neural information processing systems (NeurIPS)*, vol. 28, 2015, p. 802.
- [21] Y. Gal *et al.*, "Uncertainty in deep learning," *phd thesis, University of Cambridge*, 2016.
- [22] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *international conference on machine learning*. PMLR, 2016, pp. 1050–1059.
- [23] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," in *Advances in neural information processing systems (NeurIPS)*, vol. 30, 2017, pp. 6405–6416.
- [24] Y. Lou, Q. Song, Q. Xu, R. Tan, and J. Wang, "Uncertainty-encoded multi-modal fusion for robust object detection in autonomous driving," in *European Conference on Artificial Intelligence (ECAI)*. IOS Press, 2023, pp. 1593–1600.
- [25] D. Feng, Y. Cao, L. Rosenbaum, F. Timm, and K. Dietmayer, "Leveraging uncertainties for deep multi-modal object detection in autonomous driving," in *2020 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2020, pp. 877–884.
- [26] S. Su, Y. Li, S. He, S. Han, C. Feng, C. Ding, and F. Miao, "Uncertainty quantification of collaborative detection for self-driving," in *Proceedings of the International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 5588–5594.
- [27] S. Su, S. Han, Y. Li, Z. Zhang, C. Feng, C. Ding, and F. Miao, "Collaborative multi-object tracking with conformal uncertainty propagation," *IEEE Robotics and Automation Letters*, 2024.
- [28] A. Kendall and Y. Gal, "What uncertainties do we need in bayesian deep learning for computer vision?" in *Advances in neural information processing systems (NeurIPS)*, vol. 30, 2017, pp. 5580–5590.
- [29] D. P. Kingma, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.
- [30] Q. Chen, X. Ma, S. Tang, J. Guo, Q. Yang, and S. Fu, "F-cooper: Feature based cooperative perception for autonomous vehicle edge computing system using 3d point clouds," in *Proceedings of the ACM/IEEE Symposium on Edge Computing*, 2019, pp. 88–100.
- [31] D. Misra, "Mish: A self regularized non-monotonic activation function," *arXiv preprint arXiv:1908.08681*, 2019.
- [32] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "Carla: An open urban driving simulator," in *Proceedings of the Conference on robot learning*. PMLR, 2017, pp. 1–16.
- [33] R. Xu, Y. Guo, X. Han, X. Xia, H. Xiang, and J. Ma, "Openca: an open cooperative driving automation framework integrated with co-simulation," in *Proceedings of the IEEE International Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2021, pp. 1155–1162.
- [34] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE/CVF International conference on computer vision (ICCV)*, 2017, pp. 2980–2988.
- [35] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019.